

First Named Inventor	Jose G. Rivera	REPLY BRIEF
Serial No.	10/786,843	
Filing Date	February 25, 2004	
Group Art Unit	2192	
Examiner Name	Zheng Wei	
Confirmation No.	2936	
Attorney Docket No.	200312292-1	
Title: METHOD AND APPARATUS FOR MONITORING COMPUTER SOFTWARE		

REPLY BRIEF

Mail Stop: Appeal Brief - Patents
Commissioner for Patents
P. O. Box 1450
Alexandria, VA 22313-1450

Table of Contents

I.	Introduction.....	3
II.	Argument	4
III.	Conclusion	8

I. Introduction

An Examiner's Answer addressing Appellant's Appeal Brief was mailed on April 17, 2009.

II. Argument

Appellant responds in sections to the Examiner's Grounds of Rejection (9), and to the Examiner's Response to Argument (10), each of which continue to contain errors and are unsupportable to properly reject the claims.

With respect to the Grounds of Rejection for claim 1, the Examiner's Answer on page 4 expressly admits that Williams "does not explicitly disclose[sic] wherein the executing process is integral to an operating system." This limitation is recited by the claim. The Examiner's Answer relies on GNU for its claimed teaching of "using assert method in the operating system" (see also, Examiner's Answer, page 4). This is incorrect, and in error. The Examiner's Answer claims as support for its statement GNU's statement that GNU "check[s] for an error return from an operating system function." The Examiner's Answer contains no support in the Grounds of Rejection for this finding. Appellant submits that GNU teaches exactly that, checking for an error from an operating system, not integral to an operating system, as is recited by the claim.

"Integral to" is used in its common form in the specification, and that common form is defined as "part of" by dictionary definition. Appellant's responses to Office Actions have repeatedly used the term "integral to" in its common form, that is that the executing process is integral to (part of) the operating system. This has, in various responses, been indicated as showing that the executing process is part of the operating system, and further, that the operating system is separate from user programs, not integral thereto. See for example, application at paragraph [0012] which states in pertinent part:

"Accordingly, where a process is, for example, integral to an operating system, the operating system will not "crash the system". Likewise, where a process is integral to a user application, a user will not be frustrated as the application is commonly used."

This use of the term "integral to" shows that for the purposes of the specification, which must be used in interpretation of the claims, "integral to" is clearly different from any interpretation the Examiner's Answer attempts to substitute for the actual meaning of the term.

In the Response to Arguments section, the Examiner's Answer purports to provide additional support for the teachings of GNU being applicable to the limitation "wherein the executing process is integral to the operating system" (see Examiner's Answer, pages 12-13). However, no additional support is shown. Instead, the Examiner's Answer repeats that GNU has an assertion method "which is used to check an error return from an operating system function (integral to an operating system)" (emphasis in Examiner's Answer). This is not support. It is, instead, an allegation without any backing in fact. The Examiner's Answer continues, to apparently equate a multitude of terms with "integral to" by claiming that anything that interacts with or runs under an operating system is required to be integral to that operating system. There is no support for this assertion either. Instead, the Examiner's Answer simply equates multiple different functions with one in a convoluted and unsupported argument. Where is the support that interacting with or running under equals integral to? As has been discussed herein, integral to means part of. It is clear from a reading of the plain language of "check an error return from an operating system function" (the Examiner's Answer's emphasized phrase) that the assertion method is separate from the operating system, and is in no way integral to it, as that term should be interpreted. The external program receives an error return from the separate operating system, and acts upon it. The external program is not integral to the operating system. It is clearly a separate program. Just because a program operates with or even under another program does not mean it is integral to that program. The operating system does not need the external program to operate. It operates by itself.

The Examiner's Answer equates interaction with and integral to as the same thing. Clear definitions and meanings preclude such a construction. No reasonable interpretation allows something such as an external program that operates with or runs on an operating system to be "integral to" that operating system. Instead, it may work in conjunction with, but "integral to" as it is used precludes the construction of the Examiner's Answer. The construction in the Examiner's Answer is clearly in error.

With respect to the Examiner's Answer response to argument 2 (page 13 of Examiner's Answer), the Examiner's Answer explicitly admits that GNU is checking an error return from the operating system (see Examiner's Answer, page 13). The Examiner's Answer claims that an "executing process that issues/invokes/calls for the [assertion] has to

interact with/run under (integral to) an operating system as one of the executing processes controlled by the operating system.” This is a further attempt to integrate all external software with an operating system without any support whatsoever for so doing. External programs are certainly capable of operating without being controlled by an operating system, and operating systems are certainly capable of operating without running external programs. It appears that the Examiner’s Answer confuses what the term “integral to” actually means. It is not such a broad term that any program is integral to the operating system. This is further supported by the usage of the term in the responses to office actions, specifically those dated October 25, 2007 (amendment submitted with RCE) and April 10, 2008, as well as in the Appeal Brief.

The Examiner’s Answer response to arguments 2 section also is in error in its interpretation, without any support, of user selectable options in a representative dialog box (see page 15 of Examiner’s Answer). The Examiner’s Answer identifies three option buttons, abort, retry, and ignore, and then, without any support for so asserting, indicates that they “mean ‘Assertion Failed: Abort=Quit, Retry=Debug, Ignore=Continue’ and that is to say, the user can select or click the ‘Ignore’ button to continue executing the process.” This is only one possible interpretation of a meaning for a representative dialog box. Nowhere in the Examiner’s Answer is it indicated where abort=quit, retry=debug, and ignore=continue are taught, disclosed, or defined. In fact, Williams is completely silent as to what the terms mean. The Examiner’s Answer reads a specific meaning into terms without any support. Further, it is common knowledge that simply because a button is labeled “Retry” does not mean that a debug process is started when the button is clicked. Still further, an “Ignore” button commonly sends a user into an infinite loop, and the only way to exit is to abort the program or process. The Examiner’s Answer presumes without any support that “the user can select or click the ‘Ignore’ button to continue executing the process.”

In response to the Examiner’s Answer response to arguments section 3, the Examiner’s Answer asserts that GNU “teaches another option to continue running the program after printing its error messages” (Examiner’s Answer page 16). However, the Examiner’s Answer ignores that the claim recites both “receiving an assertion from an executing process, wherein the executing process is integral to the operating system” and “allowing the executing process to continue execution.” In contrast, it is only GNU’s

external program (i.e., not integral to the operating system) that could even be the subject of a continued execution, as the entire section of text quoted for the Examiner's Answer at page 16 is for the program that receives the assertion from an executing process, not the executing process itself. It is the executing process that sourced the assertion that is allowed to continue operating. In the case of claim 1, the executing process that sourced the assertion is the operating system. There is no teaching whatsoever in Williams or GNU of an operating system sourcing the assertion, and then continuing to run. The Examiner's Answer substitutes its own interpretation of terms and functions, ignoring the clear and unambiguous teaching of the references themselves. This is error.

Claim 1 is clearly allowable for at least the reasons set forth above, as the cited references do not teach, either alone or in any combination, each and every element of the claim. As such, claim 1 is allowable.

The remaining independent claims, rejected without further substantive argument, each contain similar elements to those of claim 1, and are also allowable for the reasons given in support of the allowance of claim 1.

The Examiner's Answer, and the previous rejections, make numerous assertions that are unreasonable in view of the plain language of the claims, the specification of the present application, Appellant's arguments, and the cited references. The Examiner's Answer arguments are unsupported except by argument alone, and are in error, as has been shown herein and in Appellant's previous responses, which are each incorporated by reference herein in their entireties.

III. Conclusion

For at least the reasons discussed above, and for reasons as presented in Appellant's Appeal Brief and previous responses, Appellant submits that all pending claims are patentable. Accordingly, Appellant requests that the Board of Appeals reverse the Examiner's decisions regarding claims 1, 4-11, 14-21, 24-31, 34-41 and 44-49.

Respectfully submitted,

Date: 15 June 2009



Daniel J. Polglaze
Reg. No. 39,801

Attorneys for Appellant
HEWLETT-PACKARD COMPANY
Intellectual Property Administration
3404 East Harmony Rd.
Fort Collins, CO 80527-2400